Class syllabus and introductions

Lecture 00

Instructor

- Julio Marco B. Pineda
- Senior in Bioengineering and Math
- Peer advisor for the Bioengineering Dept.
- Undergraduate researcher in the Pun lab
  - Cancer therapeutics
- Hobbies!
  - Photography, video games, anime, literature
  - Watching YouTube!

Contact Information

- Email: juliomp@uw.edu

- No formal office hours:
  - Feel free to send me emails if you have any questions!
  - Study sessions: Tutors and counselors

- Github repository for all files in this class:
  - https://github.com/juliomarcopineda/ClassRepository
  - Save!

Before we begin…

Introduce yourself to the class!

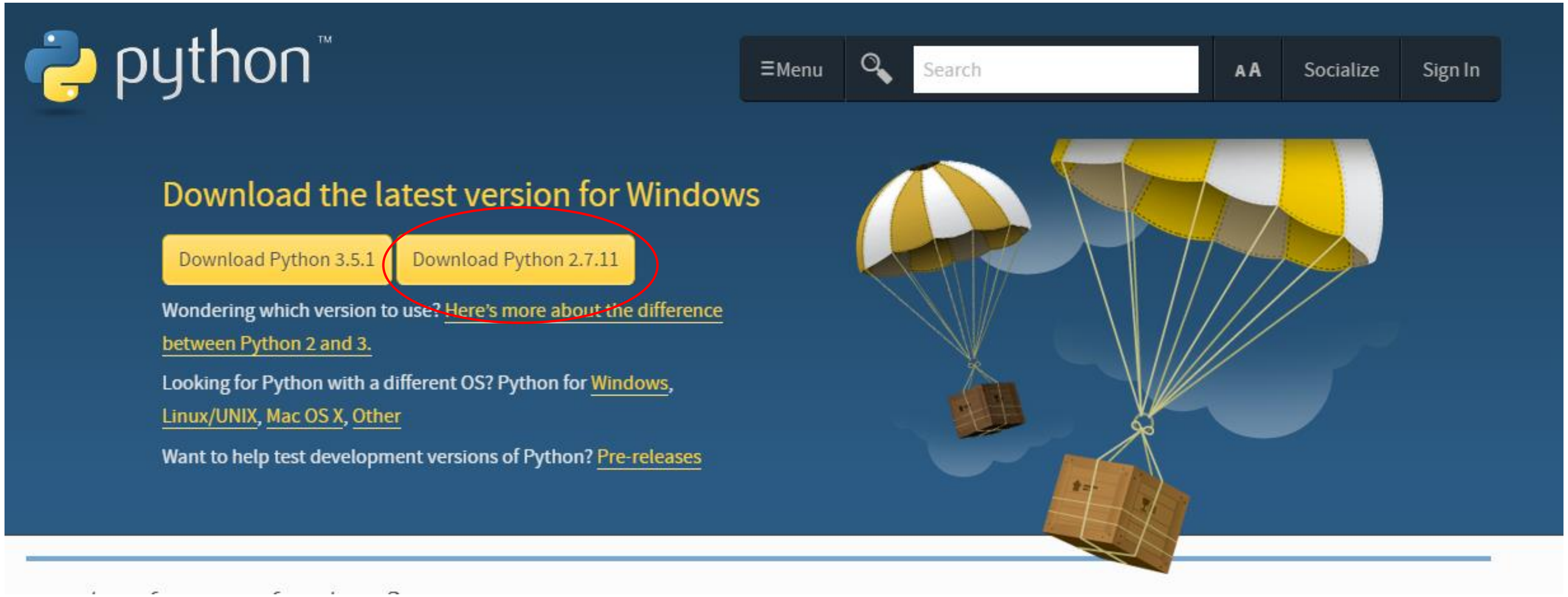
What is our class?

- Introductory Python programming course
  - Syntax and style
  - Control flow, functions, some data structures
  - Basic algorithms

- A taste of applied math/programming course
  - Homework, for the most part, will address a real-world problem
  - Final project will tackle a challenging problem

Setup time!

- Installing Python essentials:
  - https://www.python.org/downloads/



Download the latest version for Windows

Wondering which version to use? Here's more about the difference between Python 2 and 3.

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

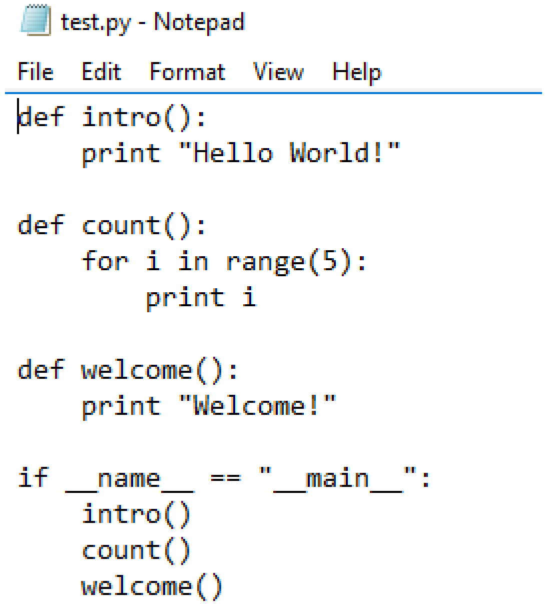
Want to help test development versions of Python? Pre-releases

Command prompt (optional)

- Use any text editing software you have:
  - Notepad, Sublime Text 3, etc.

- Use command line (cmd) to run python code

Command prompt (optional)

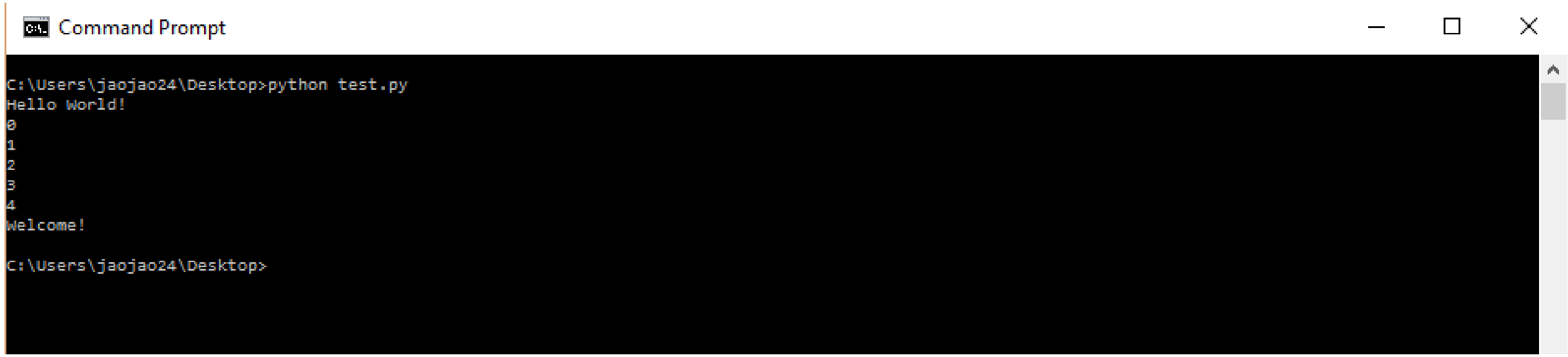


```
def intro():
    print "Hello World!"

def count():
    for i in range(5):
        print i

def welcome():
    print "Welcome!"

if __name__ == "__main__":
    intro()
    count()
    welcome()
```



```
C:\Users\jaojao24\Desktop>python test.py
Hello World!
0
1
2
3
4
Welcome!

C:\Users\jaojao24\Desktop>
```

Integrated development environment (IDE)

- Recommended for this class because it’s simple.
- Many IDE’s exist out there many have their unique strengths and weaknesses.
- Some basic features to look for:
  - Code coloring
  - Debugging
  - Interface

Suggested IDE: Spyder

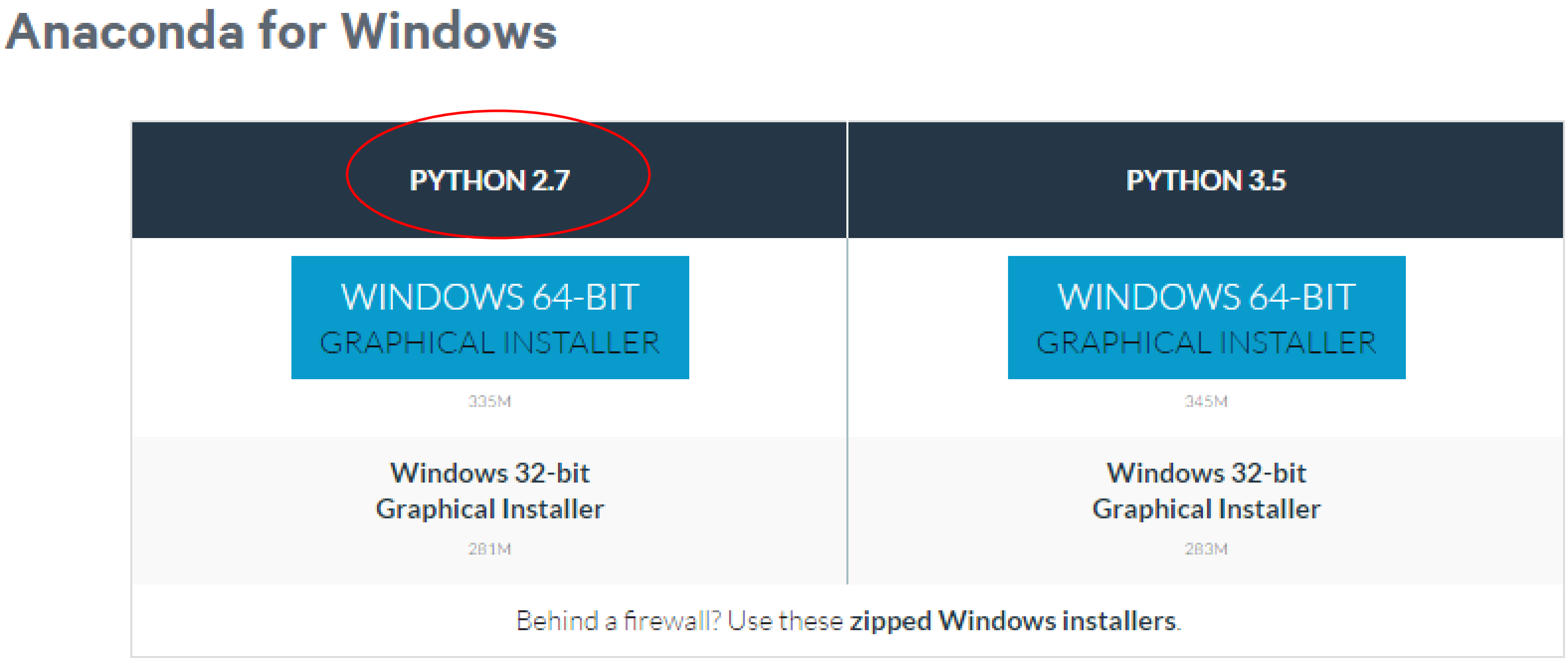
- I have used this in my bioengineering course

- Simple to follow and has the essential features

- Feel free to use any other one if you have preferences:
  - PyCharm, PyDev, etc.

Installation Instructions

- https://www.continuum.io/downloads



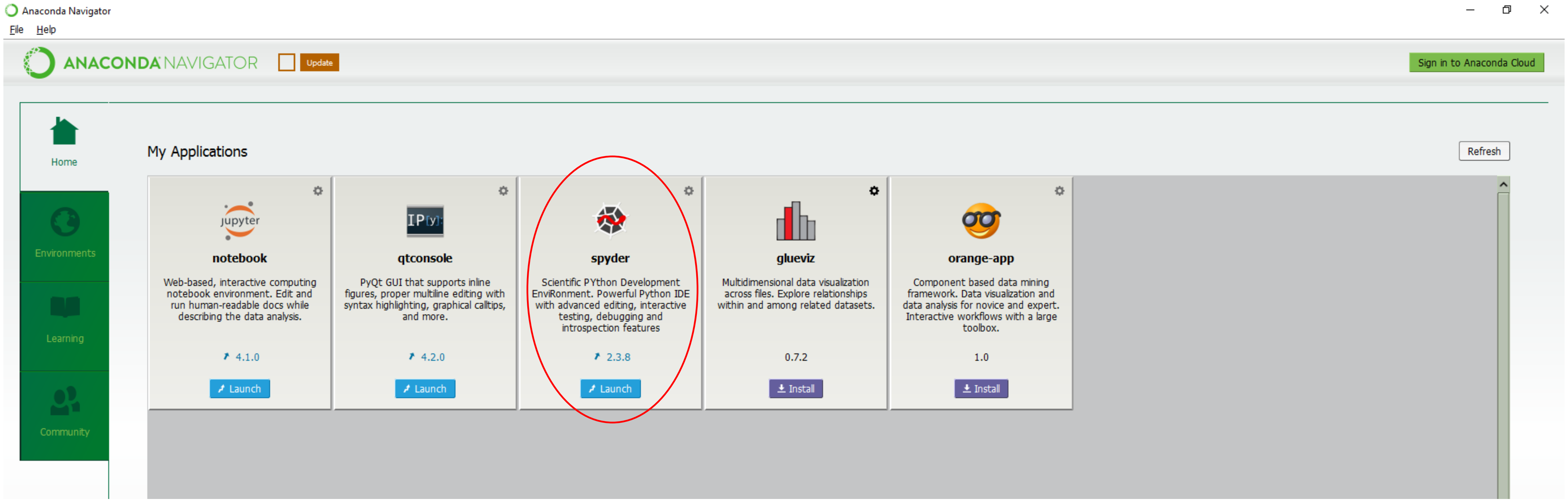
Anaconda for Windows

PYTHON 2.7	PYTHON 3.5
WINDOWS 64-BIT GRAPHICAL INSTALLER 335M	WINDOWS 64-BIT GRAPHICAL INSTALLER 345M
Windows 32-bit Graphical Installer 281M	Windows 32-bit Graphical Installer 283M

Behind a firewall? Use these **zipped Windows installers**.

Installation Instructions

- After installing by clicking the .exe file:



My Applications

notebook
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
4.1.0

qtconsole
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
4.2.0

spyder
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
2.3.8

glueviz
Multidimensional data visualization across files. Explore relationships within and among related datasets.
0.7.2

orange-app
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
1.0

Spyder Demo

Some book resources

- Think Python: How to Think Like a Computer Scientist by Allen Downey
    - Free! http://www.greenteapress.com/thinkpython/thinkpython.pdf

- Introduction to Computation and Programming Using Python: Revised and Expanded Edition by John V. Guttag
    - Not so free ☹
    - Will contact Lisa/Greg to get you copies if possible
    - I will not heavily rely on this book.

What is programming?

- A program is a sequence of instructions that specifies how to perform a computation.

- How is this different from recipes? Algorithms? Checklist?

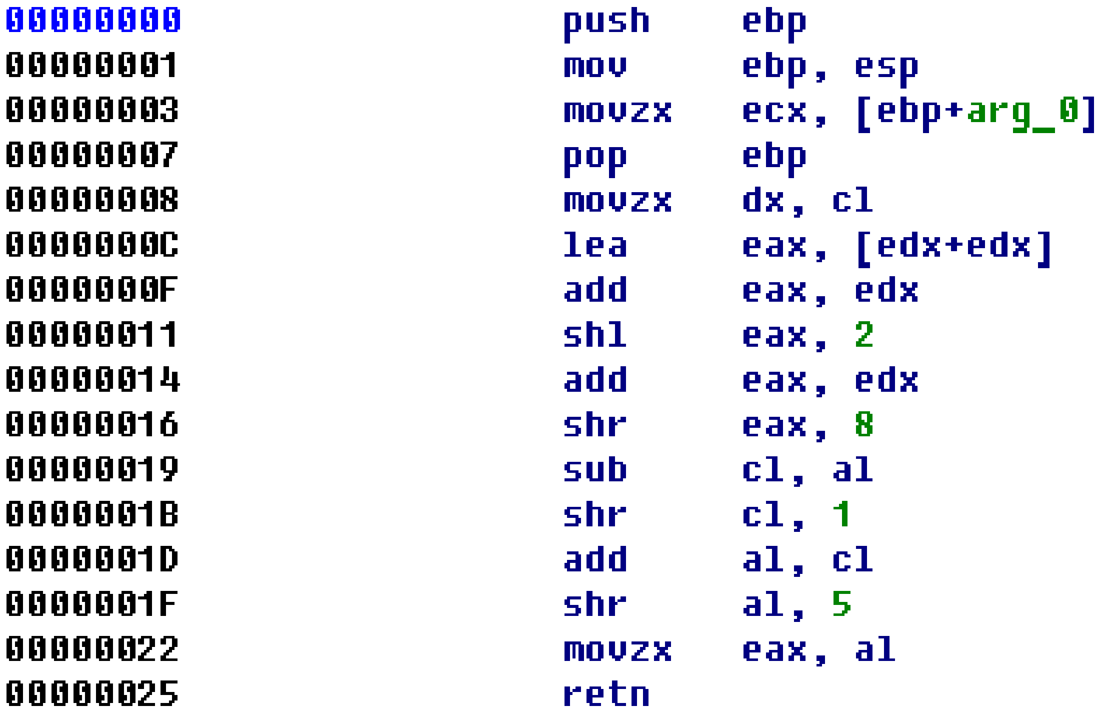
Basic properties of all programs

- **Input:** Get data from the keyboard, file, other source
- **Output:** Display data on screen
- **Math:** Perform basic math operations
- **Conditional Execution:** Check if conditions are met, then execute code
- **Repetition:** Perform some action repeatedly, with some variations

Break a large, complex task into smaller and smaller subtasks that are simple enough to be performed by these basic instructions!

High vs Low level languages

- **Low level languages:** “Machine languages” or “assembly languages”
  - Computers can only run programs written in low-level languages
  - Not all computers are created equal! Need to re-write for different types.



```
00000000                push    ebp
00000001                mov     ebp, esp
00000003                movzx   ecx, [ebp+arg_0]
00000007                pop     ebp
00000008                movzx   dx, cl
0000000C                lea     eax, [edx+edx]
0000000F                add     eax, edx
00000011                shl     eax, 2
00000014                add     eax, edx
00000016                shr     eax, 8
00000019                sub     cl, al
0000001B                shr     cl, 1
0000001D                add     al, cl
0000001F                shr     al, 5
00000022                movzx   eax, al
00000025                retn
```

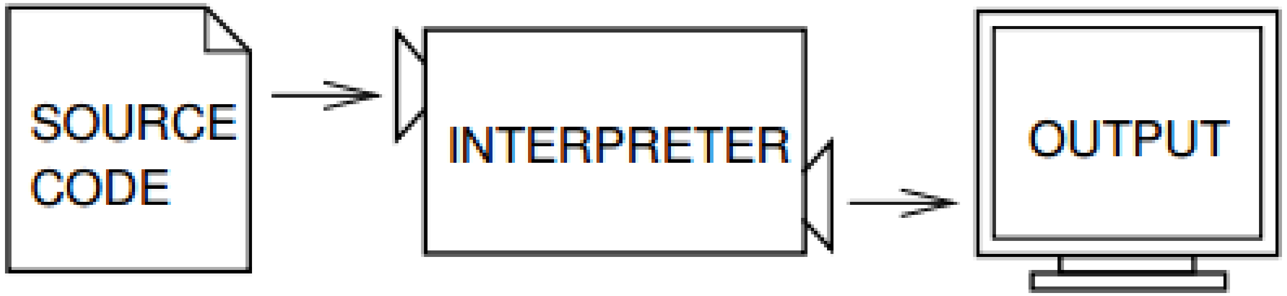
High vs Low level languages

- **High level language**
  - Easier to understand
  - Takes less time to write
  - Can be run by different types of computers

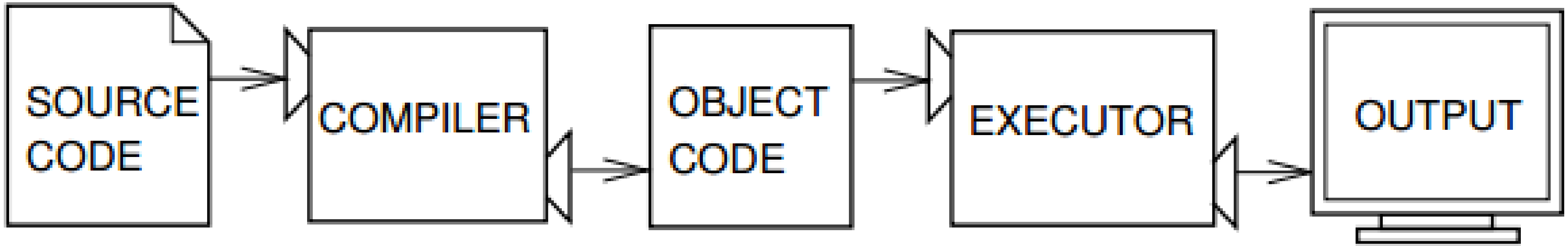
- **Is Python high or low level?**

How are programs executed

- Interpreter·



- Compiler



Why do we use Python?

- Great first programming language
  - Syntax is similar to everyday English
- You do not want to re-invent the wheel when dealing with large projects
  - Vast resources for libraries (will explain later) to use other people’s code to perform basic instructions
- Widely used by industry!

Personal opinion: The language really does not matter. As long as you understand the programming concepts! You can easily switch to any high level programming language.